

main t1	F_{ig} , 2(d)
main t1 t1.start t1.start i = print(foo) print(foo) bar =	F1g. 2(c)
main t1 [t1.start j =	Fig. 2 (b)
main t1 t1.start t1.start k = foo = foo = bar = print(foo)	time Fig. 2 (a)

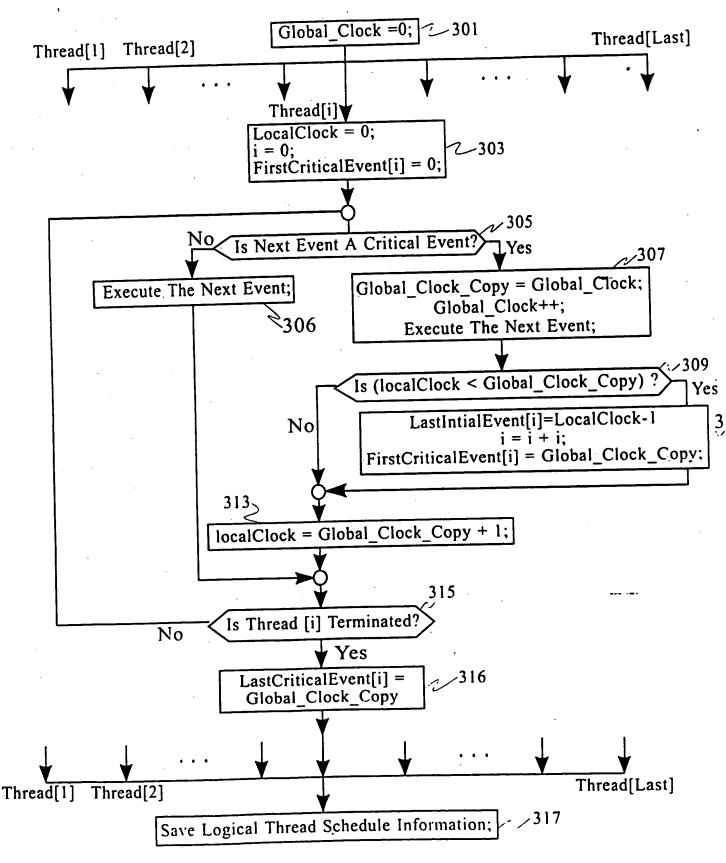
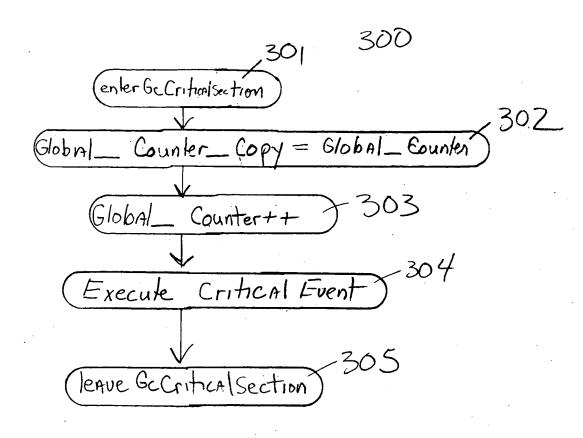
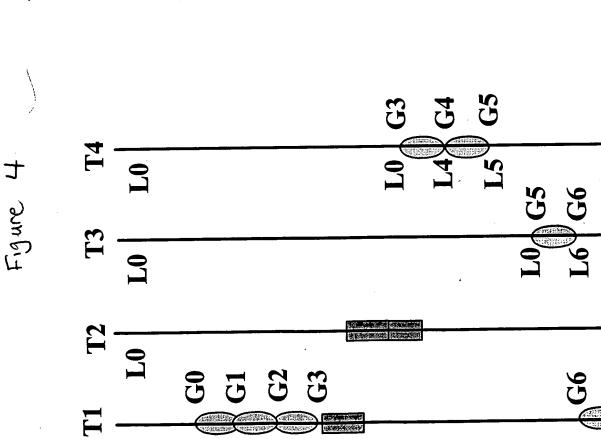


Fig. 3A



-19. 3B



L2

(: thread shared variable

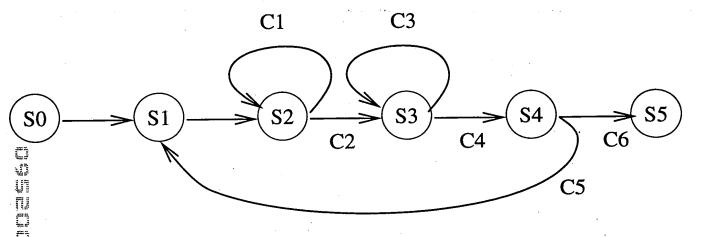
: thread local variable

Gi: global clock value Li: local clock value Thread Schedule Intervals:

T1: <0,2>, <7,8> T3: <6,6>

95_/

, **G**7



 $\mathbf{START}, \mathbf{i} = 0$

update FirstCriticalEvent(i) and

LastCriticalEvent(i)

yield the thread schedule

execute event.

CriticalEvent increment global_counter

i = i + 1

END

C1: global_counter < FirstCriticalEvent(i)

C2: not C1

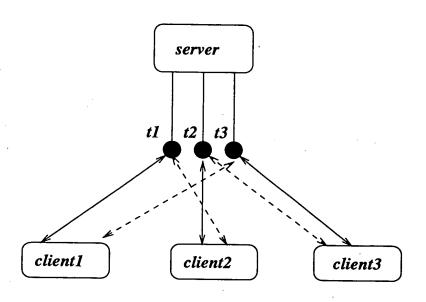
C3: global_counter <= LastCriticalEvent(

C4: not C3

C5: i <= last interval

C6: not C5

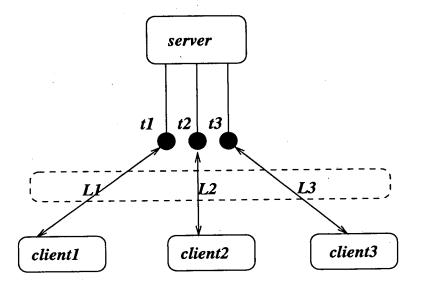
FIG. 5



connection during 1st execution

← - -> connection during 2nd execution.

TIG. 6

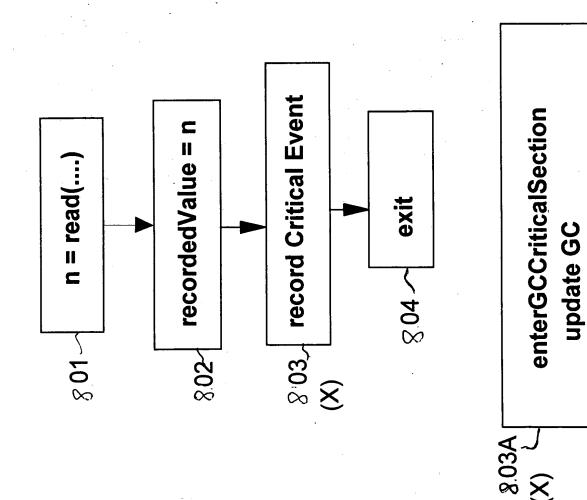


L1: <gS1,Client1Id>, Client1Id = <Client1VMID, gCounterClient1>>

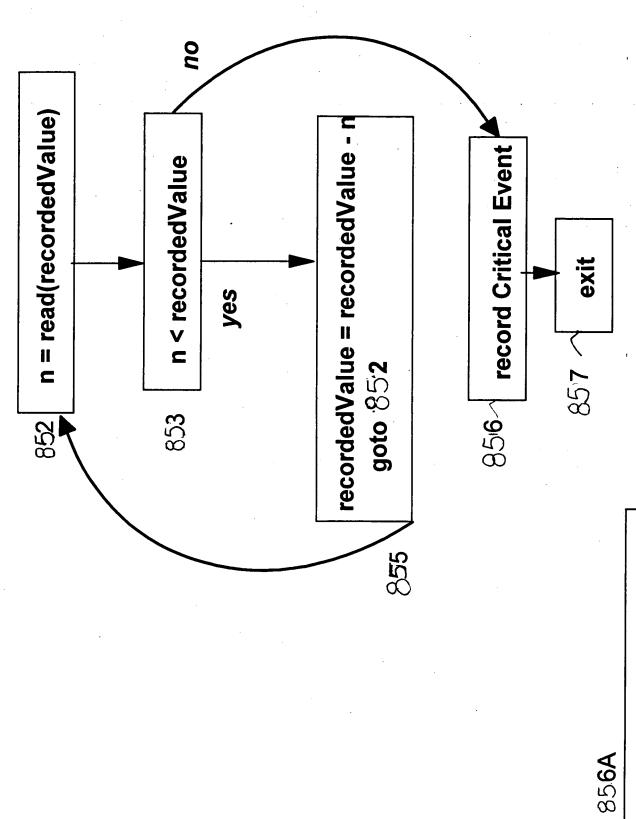
L2: <gS2, Client2Id>, Client2Id = <Client2VMID, gCounterClient2> >

L3: <gS3, Client3Id>, Client3Id = <Client3VMID, gCounterClient3>>

FIG. 7



leaveGCCriticalSection



wait until GC = recorded GC update GC

Figure 9 (a): write in Record Mode

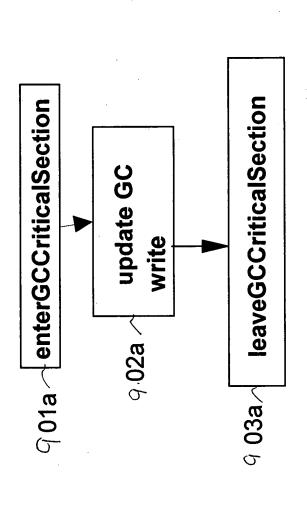


Figure 9 (b): write in Replay Mode

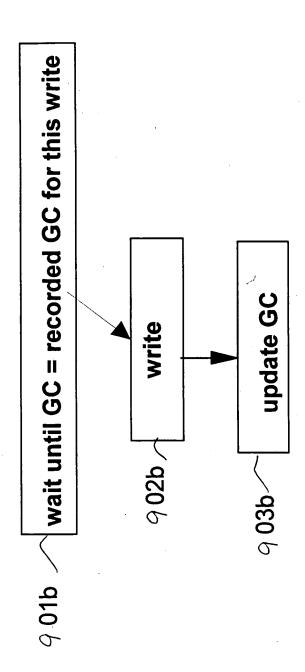


Figure 10: accept and connect in Record Mode

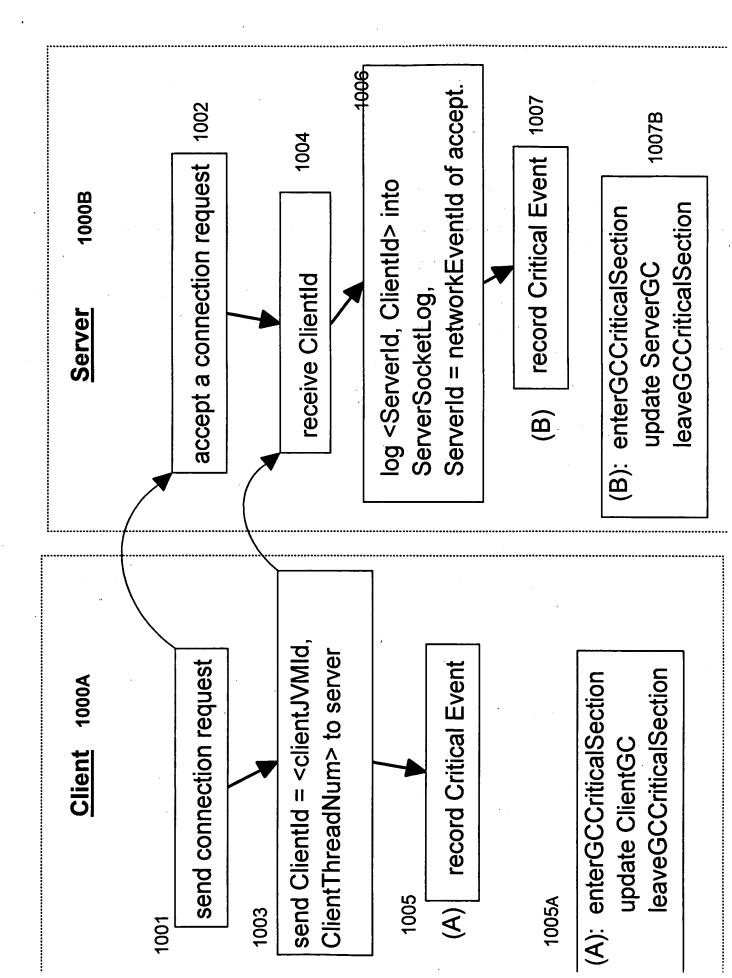
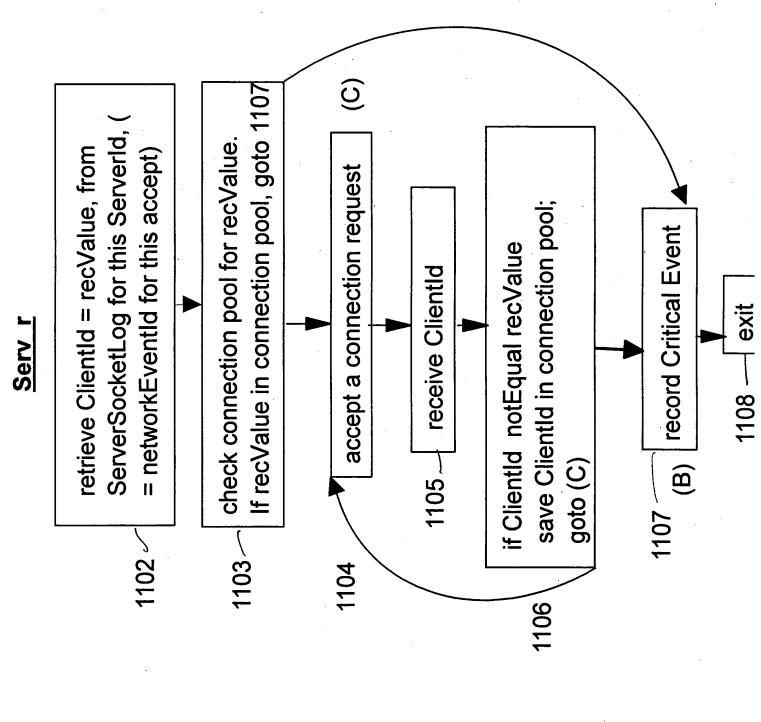


Figure 11: accept in Replay Mode



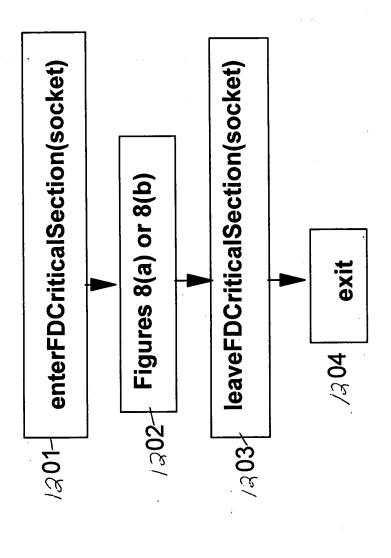


Figure 12(b): efficient replay of write

